Published since January 1970 and Published before September 2003
Terms used **address trace**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window
Try an Advanc
Try this search

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 ne
Best 200 shown

**1** ATUM: a new technique for capturing address traces using microcode
A. Agarwal, R. L. Sites, M. Horowitz
June 1986 **ACM SIGARCH Computer Architecture News , Proceeding annual international symposium on Computer architecture** 14 Issue 2
**Publisher:** IEEE Computer Society Press, ACM Press
Full text available: pdf(894.10 KB) Additional Information: full citation, abst citings, index ter

Trace-driven simulation is often used in the design of computer systems, and translation lookaside buffers. Capturing address traces to drive such been problematic, often involving 1000:1 software overhead to trace a ta and/or mechanisms that cause significant distortions in the recorded data for capturing address traces has been developed to use a processor's mici addresses in a reserved part of main memory as ...

**2** Techniques for compressing program address traces
Andrew R. Pleszkun
November 1994 **Proceedings of the 27th annual international symposiu Microarchitecture**
**Publisher:** ACM Press

Full text available: pdf(931.63 KB) Additional Information: full citation, abst citings, index ter

In this paper a technique for generating consistent, reproducible traces w of magnitude better compression than standard general-purpose compres described. With this approach, the trace is read once, an intermediate for then read as the input to the second pass over the address stream. No pro required, and this technique will work on address streams that include O of the way the address trace is encod ...

**Keywords**: compression, trace generation

3 Address trace compression through loop detection and reduction

E. N. Elnozahy

May 1999 **ACM SIGMETRICS Performance Evaluation Review , Proc 1999 ACM SIGMETRICS international conference on Meas modeling of computer systems SIGMETRICS '99**, Volume 2

**Publisher:** ACM Press

Full text available: pdf(226.94 KB) Additional Information: full citation, refe index terms

**Keywords**: address traces, compression, control flow analysis, traces

4 Generation and analysis of very long address traces

Anita Borg, R. E. Kessler, David W. Wall

May 1990 **ACM SIGARCH Computer Architecture News , Proceeding annual international symposium on Computer Architecture** 18 Issue 3a

**Publisher:** ACM Press

Full text available: pdf(1.08 MB) Additional Information: full citation, abst citings, index ter

Existing methods of generating and analyzing traces suffer from a variet including complexity, inaccuracy, short length, inflexibility, or applicabi machines. We use a trace generation mechanism based on link-time cod

which is simple to use, generates accurate long traces of multi-user progr RISC machine, and can be flexibly controlled. On-the-fly analysis of the get accurate performance data for large second-l ...

5 RATCHET: real-time address trace compression hardware for extended tra
Colleen D. Schieber, Eric E. Johnson
April 1994 **ACM SIGMETRICS Performance Evaluation Review,** Vol
**Publisher:** ACM Press
Full text available: pdf(783.24 KB) Additional Information: full citation, abst citings, index ter

The address traces used in computer architecture research are commonly software techniques that introduce time dilations of an order of magnitud techniques may also omit classes of memory references that are importa models of computer systems, such as instruction prefetches, operating sy and interrupt activity.We describe a technique for capturing all classes o time. RATCHET employs trace filtering hardware to redu ...

6 Constructing instruction traces from cache-filtered address traces (CITCAT
Charlton D. Rose, J. Kelly Flanagan
December 1996 **ACM SIGARCH Computer Architecture News,** Volun
**Publisher:** ACM Press
Full text available: pdf(595.54 KB) Additional Information: full citation, abst terms

Instruction traces are useful tools for studying many aspects of computer are difficult to gather without perturbing the systems being traced. In the have collected instruction traces through various techniques, including si instruction inlining, hardware monitoring, and processor simulation. The however, fail to produce accurate traces because they interfere with the p execution.Because processors are deterministic ...

7 TRAPEDS: producing traces for multicomputers via execution driven simu
C. B. Stunkel, W. K. Fuchs
April 1989 **ACM SIGMETRICS Performance Evaluation Review , Pro 1989 ACM SIGMETRICS international conference on Mea modeling of computer systems SIGMETRICS '89,** Volume
**Publisher:** ACM Press

Full text available: pdf(960.90 KB) Additional Information: full citation, abst citings, index ter

Trace-driven simulation is an important aid in performance analysis of c Capturing address traces for these simulations is a difficult problem for s and particularly for multicomputers. Even when existing trace methods c multicomputers, the amount of collected data typically grows with the nu so I/O and trace storage costs increase. A new technique is presented in t modifies the executable code to dynamically col ...

8 Session 9: traffic analysis: Observed structure of addresses in IP traffic
Eddie Kohler, Jinyang Li, Vern Paxson, Scott Shenker
November 2002 **Proceedings of the 2nd ACM SIGCOMM Workshop o measurment**
**Publisher:** ACM Press
Full text available: pdf(1.18 MB) Additional Information: full citation, abst citings, index ter

This paper investigates the structure of addresses contained in IP traffic. analyze the structural characteristics of destination IP addresses seen on considered as a subset of the address space. These characteristics may ha algorithms that deal with IP address aggregates, such as routing lookups congestion control. We find that address structures are well modeled by Cantor dust with two parameters. The model m ...

9 Trace-driven memory simulation: a survey
Richard A. Uhlig, Trevor N. Mudge
June 1997 **ACM Computing Surveys (CSUR)**, Volume 29 Issue 2
**Publisher:** ACM Press
Full text available: pdf(636.11 KB) Additional Information: full citation, abst citings, index ter

As the gap between processor and memory speeds continues to widen, m evaluating memory system designs before they are implemented in hard increasingly important. One such method, trace-driven memory simulati subject of intense interest among researchers and has, as a result, enjoye and substantial improvements during the past decade. This article survey developments by establishing criteria for evaluating trac ...

**Keywords**: TLBs, caches, memory management, memory simulation, tr simulation

**10** Execution-driven simulation of multiprocessors: address and timing analys
S. Dwarkadas, J. R. Jump, J. B. Sinclair
October 1994 **ACM Transactions on Modeling and Computer Simulati** Volume 4 Issue 4
**Publisher:** ACM Press
Full text available: pdf(1.58 MB) Additional Information: full citation, abst citings, index ter

This article describes and evaluates an efficient execution-driven techniq simulation of multiprocessors that includes the simulation of system mer driven by real program work loads. The technique produces correctly int traces at run-time without disk access overhead or hardware support, allc simulation of the effects of a variety of architectural alternatives on prog implemented a simulator based on this technique that offe ...

**Keywords**: distributed systems, execution-driven simulation, parallel tra memory multiprocessors

**11** Designing a trace format for heap allocation events
Trishul Chilimbi, Richard Jones, Benjamin Zorn
October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd interna on Memory management ISMM '00,** Volume 36 Issue 1
**Publisher:** ACM Press
Full text available: pdf(1.53 MB) Additional Information: full citation, abst terms

Dynamic storage allocation continues to play an important role in the pe correctness of systems ranging from user productivity software to high-p While algorithms for dynamic storage allocation have been studied for d literature is based on measuring the performance of benchmark program of many important allocation-intensive workloads. Furthermore, to date emerged or been proposed for publishing and exchangin ...

**12** Dynamic base register caching: a technique for reducing address bus width
Matthew Farrens, Arvin Park
April 1991 **ACM SIGARCH Computer Architecture News , Proceedin annual international symposium on Computer architecture** 19 Issue 3
**Publisher:** ACM Press
Full text available: pdf(948.04 KB) Additional Information: full citation, refe index terms

**13** Dealing with high speed links and other measurement challenges: On the d performance of prefix-preserving IP traffic trace anonymization
Jun Xu, Jinliang Fan, Mostafa Ammar, Sue B. Moon
November 2001 **Proceedings of the 1st ACM SIGCOMM Workshop on Measurement**
**Publisher:** ACM Press
Full text available: pdf(697.42 KB) Additional Information: full citation, refe index terms

**14** An in-cache address translation mechanism
D. A. Wood, S. J. Eggers, G. Gibson, M. D. Hill, J. M. Pendleton
June 1986 **ACM SIGARCH Computer Architecture News , Proceeding annual international symposium on Computer architecture** 14 Issue 2
**Publisher:** IEEE Computer Society Press, ACM Press
Full text available: pdf(770.30 KB) Additional Information: full citation, abst citings, index ter

In the design of SPUR, a high-performance multiprocessor workstation, caches and hardware-supported cache consistency suggests a new appro address translation. By performing translation in each processor's virtual need for separate translation lookaside buffers (TLBs) is eliminated. Elir substantially reduces the hardware cost and complexity of the translation eliminates the translation consistency problem. Trac ...

**15** Optimal tracing and incremental reexecution for debugging long-running p

Robert H. B. Netzer, Mark H. Weaver
June 1994 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLA on Programming language design and implementation PLDI** Issue 6
**Publisher:** ACM Press
Full text available: pdf(1.34 MB) Additional Information: full citation, refe index terms

**16** Memory-wall: Boosting trace cache performance with nonhead miss specu
Stevan Vlaovic, Edward S. Davidson
June 2002 **Proceedings of the 16th international conference on Superco**
**Publisher:** ACM Press
Full text available: pdf(179.52 KB) Additional Information: full citation, abst index terms

Trace caches are used to help dynamic branch prediction make multiple cycle by embedding some of the predictions in the trace. In this work, w cache that is capable of delivering a trace consisting of a variable numbe a linked list mechanism. We evaluate several schemes in the context of a model that stores decoded instructions. By developing a new classificati accesses, we are able to target those misses t ...

**Keywords**: branch prediction, optimization, trace cache, x86

**17** Trace cache: a low latency approach to high bandwidth instruction fetchin
Eric Rotenberg, Steve Bennett, James E. Smith
December 1996 **Proceedings of the 29th annual ACM/IEEE internation Microarchitecture**
**Publisher:** IEEE Computer Society
Full text available: pdf(1.38 MB) Additional Information: full citation, abst citings, index ter

As the issue width of superscalar processors is increased, instruction fetc requirements will also increase. It will become necessary to fetch multip cycle. Conventional instruction caches hinder this effort because long in are not always in contiguous cache locations. We propose supplementin

instruction cache with a trace cache. This structure caches traces of the d stream, so instructions that are otherwise no ...

**Keywords**: instruction cache, instruction fetching, multiple branch predi processors, trace cache

**18** Address compression through base register caching
Arvin Park, Matthew Farrens
November 1990 **Proceedings of the 23rd annual workshop and symposi Microprogramming and microarchitecture**
**Publisher:** IEEE Computer Society Press
Full text available: pdf(689.60 KB) Additional Information: full citation, abst citings

This paper presents a technique to reduce processor-to-memory address l exploiting temporal and spatial locality in address reference streams. Hig of address words are cached in base registers at both the processor and m it possible to transmit small register indexes between processor and mem high order address bits themselves. Trace driven simulations indicate tha Caching reduces processor-to-me ...

**Keywords**: CPU performance, bandwidth, locality, microprocessor syste

**19** Mache: no-loss trace compaction
A. D. Samples
April 1989 **ACM SIGMETRICS Performance Evaluation Review , Pro 1989 ACM SIGMETRICS international conference on Mea modeling of computer systems SIGMETRICS '89**, Volume
**Publisher:** ACM Press
Full text available: pdf(798.23 KB) Additional Information: full citation, abst citings, index ter

Execution traces can be significantly compressed using their referencing observation leads to a technique capable of compressing execution trace magnitude; instruction-only traces are compressed by two orders of mag technique is unlike previously reported trace compression techniques in without loss of information and, therefore, does not affect trace-driven si

accuracy.

**20** Techniques for efficient inline tracing on a shared-memory multiprocessor
S. J. Eggers, David R. Keppel, Eric J. Koldinger, Henry M. Levy
April 1990 **ACM SIGMETRICS Performance Evaluation Review , Pro 1990 ACM SIGMETRICS conference on Measurement and computer systems SIGMETRICS '90**, Volume 18 Issue 1
**Publisher:** ACM Press
Full text available: pdf(1.12 MB) Additional Information: full citation, abst citings, index ter

While much current research concerns multiprocessor design, few traces programs are available for analyzing the effect of design trade-offs. Exis methods have serious drawbacks: trap-driven methods often slow down by more than 1000 times, significantly perturbing program behavior; mi modification is faster, but the technique is neither general nor portable. T a new tool, called MPTRACE, for collecting tr ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 1

Published since January 1970 and Published before September 2003 F

Terms used **register address trace** or **instrument**

Sort results by relevance Save results to a Binder Try an Advanc
Search Tips Try this search
Display results expanded form ☐ Open results in a new window

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 ne
Best 200 shown R

**1** Optimally profiling and tracing programs
Thomas Ball, James R. Larus
July 1994 **ACM Transactions on Programming Languages and System**
Volume 16 Issue 4
**Publisher:** ACM Press
Full text available: pdf(2.84 MB) Additional Information: full citation, abst
citings, index ter

This paper describes algorithms for inserting monitoring code to profile : These algorithms greatly reduce the cost of measuring programs with res commonly used technique of placing code in each basic block. Program number of times each basic block in a program executes. Instruction trac sequence of basic blocks traversed in a program execution. The algorithr placement of counting/tracing code with respect to the ...

**Keywords**: control-flow graph, instruction tracing, instrumentation, pro

**2** Address calculation for retargetable compilation and exploration of instruc architectures
Clifford Liem, Pierre Paulin, Ahmed Jerraya

June 1996 **Proceedings of the 33rd annual conference on Design autom**
**Publisher:** ACM Press
Full text available: pdf(54.16 KB) Additional Information: full citation, refe index terms

3 Performance debugging shared memory parallel programs using run-time c
Ramakrishnan Rajamony, Alan L. Cox
June 1997 **ACM SIGMETRICS Performance Evaluation Review , Proc 1997 ACM SIGMETRICS international conference on Meas modeling of computer systems SIGMETRICS '97**, Volume 2
**Publisher:** ACM Press
Full text available: pdf(2.37 MB) Additional Information: full citation, abst citings, index ter

We describe a new approach to performance debugging that focuses on a identifying computation transformations to reduce synchronization and c grouping writes together into *equivalence classes*, we are able to tractabl information from long-running programs. Our performance debugger an information and suggests computation transformations in terms of the so present the transformations suggested by the debugger on a suite of four

4 The flight recorder: an architectural aid for system monitoring
Michael M. Gorlick
December 1991 **ACM SIGPLAN Notices , Proceedings of the 1991 ACM on Parallel and distributed debugging PADD '91**, Volu
**Publisher:** ACM Press
Full text available: pdf(944.95 KB) Additional Information: full citation, refe index terms

5 Active memory: a new abstraction for memory system simulation
Alvin R. Lebeck, David A. Wood
January 1997 **ACM Transactions on Modeling and Computer Simulati** Volume 7 Issue 1
**Publisher:** ACM Press
Full text available: pdf(690.38 Additional Information: full citation, refe

KB) index terms

**Keywords**: Cache memory, direct-execution simulation, memory hierar simulation, trace-driven simulation

6 The performance enhancement of descriptor-based virtual memory system associative registers
R. E. Brundage, A. P. Batson
December 1974 **ACM SIGARCH Computer Architecture News , Proce annual symposium on Computer architecture ISCA '7**
**Publisher:** ACM Press
Full text available: pdf(539.04 KB) Additional Information: full citation, abst citings

Contemporary paged virtual memory systems often use associative regis time to frequently-referenced pages. Here we examine the analogous use registers in descriptorbased, symbolically-segmented virtual memory sy segment contains an entire data structure as defined in a high-level langu data from production Algol 60 programs were used to determine perform as a function of the number of associative registers in ...

7 Active memory: a new abstraction for memory-system simulation
Alvin R. Lebeck, David A. Wood
May 1995 **ACM SIGMETRICS Performance Evaluation Review , Pro 1995 ACM SIGMETRICS joint international conference on modeling of computer systems SIGMETRICS '95/PERFOR**
Volume 23 Issue 1
**Publisher:** ACM Press
Full text available: pdf(1.28 MB) Additional Information: full citation, abst citings, index ter

This paper describes the *active memory* abstraction for memory-system s abstraction---designed specifically for on-the-fly simulation, memory re invoke a user-specified function depending upon the reference's type and block state. Active memory allows simulator writers to specify the appro each reference, including "no action" for the common case of cache hits.

abstraction hides implementation details, implemen ...

**8** Experience with a software-defined machine architecture
David W. Wall
May 1992 **ACM Transactions on Programming Languages and System**
Volume 14 Issue 3
**Publisher:** ACM Press
Full text available: pdf(2.86 MB) Additional Information: full citation, abst citings, index ter

We have built a system in which the compiler back end and the linker w present an abstract machine at a considerably higher level than the actua intermediate language translated by the back end is the target language o compilers and is also the only assembly language generally available. Th intermodule register allocation, which would be harder if some of the co had come from a traditional assembler, out of sight of ...

**Keywords**: RISC, graph coloring, intermediate language, interprocedura pipeline scheduling, profiling, register allocation, register windows

**9** EEL: machine-independent executable editing
James R. Larus, Eric Schnarr
June 1995 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLA on Programming language design and implementation PLD**
Issue 6
**Publisher:** ACM Press
Full text available: pdf(1.15 MB) Additional Information: full citation, abst citings, index ter

EEL (Executable Editing Library) is a library for building tools to analyz executable (compiled) program. The systems and languages communitie tools for error detection, fault isolation, architecture translation, perform simulation, and optimization using this approach of modifying executabl however, tools of this sort are difficult and time-consuming to write and tied to a particular machine and operating sy ...

**10** Constructing instruction traces from cache-filtered address traces (CITCA

Charlton D. Rose, J. Kelly Flanagan
December 1996 **ACM SIGARCH Computer Architecture News,** Volun
**Publisher:** ACM Press
Full text available: pdf(595.54 KB) Additional Information: full citation, abst terms

Instruction traces are useful tools for studying many aspects of computer are difficult to gather without perturbing the systems being traced. In the have collected instruction traces through various techniques, including si instruction inlining, hardware monitoring, and processor simulation. The however, fail to produce accurate traces because they interfere with the p execution.Because processors are deterministic ...

**11** Techniques for efficient inline tracing on a shared-memory multiprocessor
S. J. Eggers, David R. Keppel, Eric J. Koldinger, Henry M. Levy
April 1990 **ACM SIGMETRICS Performance Evaluation Review , Pro 1990 ACM SIGMETRICS conference on Measurement and computer systems SIGMETRICS '90,** Volume 18 Issue 1
**Publisher:** ACM Press
Full text available: pdf(1.12 MB) Additional Information: full citation, abst citings, index ter

While much current research concerns multiprocessor design, few traces programs are available for analyzing the effect of design trade-offs. Exis methods have serious drawbacks: trap-driven methods often slow down by more than 1000 times, significantly perturbing program behavior; mi modification is faster, but the technique is neither general nor portable. T a new tool, called MPTRACE, for collecting tr ...

**12** Measuring limits of parallelism and characterizing its vulnerability to reso
Lawrence Rauchwerger, Pradeep K. Dubey, Ravi Nair
December 1993 **Proceedings of the 26th annual international symposiu Microarchitecture**
**Publisher:** IEEE Computer Society Press
Full text available: pdf(1.39 MB) Additional Information: full citation, refe

**13** Fast instruction cache performance evaluation using compile-time analysis
David B. Whalley
June 1992 **ACM SIGMETRICS Performance Evaluation Review , Proc 1992 ACM SIGMETRICS joint international conference on modeling of computer systems SIGMETRICS '92/PERFOR** Volume 20 Issue 1
**Publisher:** ACM Press
Full text available: pdf(1.08 MB) Additional Information: full citation, refe index terms

**Keywords**: cache simulation, instruction cache, trace analysis, trace gen

**14** Automatic and efficient evaluation of memory hierarchies for embedded sy
Santosh G. Abraham, Scott A. Mahlke
November 1999 **Proceedings of the 32nd annual ACM/IEEE internatio Microarchitecture**
**Publisher:** IEEE Computer Society
Full text available: pdf(1.44 MB) Publisher Site Additional Information: full citation, abst citings, index ter

Automation is the key to the design of future embedded systems as it per specific customization while keeping design costs low. A key problem fa design systems is evaluating the performance of the vast number of alter timely manner. For this paper, we focus on an embedded system consisti components: a VLIW processor, instruction cache, data cache, and secor cache. A hierarchical approach of partitioning the ...

**15** Performance optimization of pipelined primary cache
Kunle Olukotun, Trevor Mudge, Richard Brown
April 1992 **ACM SIGARCH Computer Architecture News , Proceedin annual international symposium on Computer architecture** 20 Issue 2
**Publisher:** ACM Press

Full text available: pdf(1.11 MB) Additional Information: full citation, abst citings, index ter

The CPU cycle time of a high-performance processor is usually determi time of the primary cache. As processors speeds increase, designers will number of pipeline stages used to fetch data from the cache in order to r dependence of CPU cycle time on cache access time. This paper studies advantages of a pipelined cache for a GaAs implementation of the MIPS using a design methodology that includes long traces of ...

**16** Optimal tracing and incremental reexecution for debugging long-running p

Robert H. B. Netzer, Mark H. Weaver

June 1994 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLA on Programming language design and implementation PLD** Issue 6

**Publisher:** ACM Press

Full text available: pdf(1.34 MB) Additional Information: full citation, refe index terms

**17** Speculative execution via address prediction and data prefetching

José González, Antonio González

July 1997 **Proceedings of the 11th international conference on Superco**

**Publisher:** ACM Press

Full text available: pdf(1.33 MB) Additional Information: full citation, refe index terms

**18** Eliminating the address translation bottleneck for physical address cache

Tzi-cker Chiueh, Randy H. Katz

September 1992 **ACM SIGPLAN Notices , Proceedings of the fifth inte conference on Architectural support for programming operating systems ASPLOS-V**, Volume 27 Issue 9

**Publisher:** ACM Press

Full text available: pdf(1.28 MB) Additional Information: full citation, refe index terms

**19** Automatic incremental state saving
Darrin West, Kiran Panesar
July 1996 **ACM SIGSIM Simulation Digest , Proceedings of the tenth v Parallel and distributed simulation PADS '96,** Volume 26 Iss
**Publisher:** IEEE Computer Society, ACM Press
Full text available: pdf(870.62 KB) Publisher Site Additional Information: full citation, abst citings, index ter

We present an Incremental State Saving technique for which the state sa inserted automatically by directly editing the application executable. Thi advantage of being easy to use since it is fully automatic, and has good p adds overhead only where state is being modified. Since the editing happ code, the method is independent of the compiler, and allows third party l None of the previous incremental state saving ...

**Keywords**: Parallel Discrete Event Simulation, State Saving, Incrementa Checkpointing, Time Warp

**20** SIGMA: a simulator infrastructure to guide memory analysis
Luiz DeRose, K. Ekanadham, Jeffrey K. Hollingsworth, Simone Sbaraglia
November 2002 **Proceedings of the 2002 ACM/IEEE conference on Sup**
**Publisher:** IEEE Computer Society Press
Full text available: pdf(333.53 KB) Additional Information: full citation, abst citings, index ter

In this paper we present SIGMA (Simulation Infrastructure to Guide Me new data collection framework and family of cache analysis tools. The S provides detailed cache information by gathering memory reference data based instrumentation. This infrastructure can facilitate quick probing in influence the performance of an application by highlighting bottleneck s excessive cache/TLB misses and inefficient data layou ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 1

Sign in

Google Web Images Groups News Froogle Maps more

address trace order of execution Search Ac Pr

**Web** Results **1** - **10** of about **5,550,000** for **address trace order of execution**.

Reference - Contents
Search **Order** · Errors during **Execution** · Return Values · Built-in Functions · ABBREV (Abbreviation) · ABS (Absolute Value) · **ADDRESS** · ARG (Argument) · BEEP **...**
publib.boulder.ibm.com/infocenter/iadthelp/v6r0/topic/com.ibm.etools.iseries.orxw.doc/orxw_ref02.htm - 60k - Cached - Similar pages

**[DOC] Trace** Cache
File Format: Microsoft Word - View as HTML
Rotenberg et al [1] proposed **Trace** Cache technique to **address** the above **...** Since the instructions are stored in **execution order** in the **Trace** Cache, **...**
www.cs.ucf.edu/~mali/TraceCache.doc - Similar pages

MBR IT/.NET 247 : **order of execution** of page_load in a base and **...**
is this also the **order of execution** in inheritence or only with delegated event handlers? **...** "zf" <Click here to reveal e-mail **address**> wrote in message **...**
www.dotnet247.com/247reference/msgs/58/291438.aspx - 37k - Cached - Similar pages

calvin2+DICE and LiKE
In **order** to evaluate **execution** slowdowns incurred by both **execution** modes, **...** and data **address trace** generation (to /dev/null) in emulation mode is 117.33. **...**
www.irisa.fr/caps/projects/calvin2DICE/index.htm - 19k - Cached - Similar pages

Citations: Abstract **execution**: A technique for efficiently tracing **...**
The **trace** generator delivers **address** traces and a set of synchronization events to **...** the program in **order** to record dynamic events during its **execution**. **...**
citeseer.ist.psu.edu/context/45636/0 - 30k - Cached - Similar pages

**[PDF]** MATCH: Memory **Address Trace** CacHe
File Format: PDF/Adobe Acrobat - View as HTML
have all been examined in **order** to supply the **execution**. core with the necessary, **...** behind our memory **address trace** cache and discuss our. hypothesis. **...**
www.owlnet.rice.edu/~elec525/projects/match_report.pdf - Similar pages

DRAMsim: University of Maryland Memory-System Simulation Framework
It can also be driven by an **address trace**, taking timing and **address** information from the **...** The Effects of Out-of-**Order Execution** on the Memory System. **...**
www.enee.umd.edu/dramsim/ - 31k - Cached - Similar pages

**[PDF]** TraceVis: An **Execution Trace** Visualization Tool
File Format: PDF/Adobe Acrobat - View as HTML
ing, out-of-**order execution**, memory hierarchies and prefetch- **...** To **address** this need, we have developed TraceVis, a flexible **...**
www-faculty.cs.uiuc.edu/ ~zilles/papers/**trace**vis.mobs.pdf - Similar pages

**[RTF]** Computer-Based Investigation
File Format: Rich Text Format - View as HTML
A. Enter an **order** authorizing the installation and use of a trap and **trace** device to identify the source **address** of electronic mail communications directed **...**
www.fjc.gov/public/pdf.nsf/ lookup/CompIn02.rtf/$file/CompIn02.rtf - Similar pages

EETimes.com - Data **trace** format facilitates debugs
Additionally, the visibility to the **address** bus is unrestricted with a **...** data value by starting to work on the instructions in reverse **order of execution**, **...**
www.eet.com/story/OEG20000328S0011 - 70k - Cached - Similar pages

Try your search again on Google Book Search

Goooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

IEEE Xplore RELEASE 2.1

Welcome United States Patent and Trademark Office

Search Results BROWSE SEARCH IEEE GUID

Results for "(((address trace)<in>metadata)) <and> (pyr >= 1970 <and> 2003)"
Your search matched **41** of **1372086** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevanc Descending** order.

» Search Options

View Session History
New Search

Modify Search

(((address trace)<in>metadata)) <and> (pyr >= 197(

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

view selected items Select All Deselect All

» Key

IEEE JNL IEEE Journal or Magazine

IEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

☐ 1. **Reconfigurable real-time address trace microprocessors**
Shyh-Ming Huang; Ing-Jer Huang; Chung
Field-Programmable Technology (FPT), 2
International Conference on
15-17 Dec. 2003 Page(s):196 - 203
Digital Object Identifier 10.1109/FPT.200
AbstractPlus | Full Text: PDF(654 KB)
Rights and Permissions

☐ 2. **Generating dynamically scheduled men**
Oliver, R.L.; Teller, P.J.;
Performance, Computing and Communica
International
16-18 Feb. 1998 Page(s):245 - 250
Digital Object Identifier 10.1109/PCCC.1
AbstractPlus | Full Text: PDF(664 KB)
Rights and Permissions

☐ 3. **Compressing address traces with RECI**
Ahola, J.;
Workload Characterization, 2001. WWC-
Workshop on
2 Dec. 2001 Page(s):120 - 126
AbstractPlus | Full Text: PDF(556 KB)
Rights and Permissions

☐ 4. **Collecting address traces from parallel**
Stunkel, C.B.; Janssens, B.; Fuchs, W.K.;

System Sciences, 1991. Proceedings of th
International Conference on
Volume 1, 8-11 Jan. 1991 Page(s):373 - 3
Digital Object Identifier 10.1109/HICSS.1
AbstractPlus | Full Text: PDF(876 KB)
Rights and Permissions

☐ **5. Performance of a RISC machine with t**
Happel, L.P.; Jayasumana, A.P.;
Computers and Digital Techniques, IEE P
Volume 139, Issue 3, May 1992 Page(s):
AbstractPlus | Full Text: PDF(584 KB)

☐ **6. Compressing address trace data for cac**
Fox, A.; Grun, T.;
Data Compression Conference, 1997. DC
25-27 March 1997 Page(s):439
Digital Object Identifier 10.1109/DCC.19
AbstractPlus | Full Text: PDF(60 KB) **I**
Rights and Permissions

☐ **7. Address tracing for parallel machines**
Stunkel, C.B.; Janssens, B.; Fuchs, W.K.;
Computer
Volume 24, Issue 1, Jan. 1991 Page(s):3
Digital Object Identifier 10.1109/2.67191
AbstractPlus | Full Text: PDF(628 KB)
Rights and Permissions

☐ **8. A software approach to multiprocessor**
Azimi, M.; Erickson, C.;
Computer Software and Applications Con
Proceedings., Fourteenth Annual Internati
31 Oct.-2 Nov. 1990 Page(s):99 - 105
Digital Object Identifier 10.1109/CMPSA
AbstractPlus | Full Text: PDF(532 KB)
Rights and Permissions

☐ **9. Exploiting streams in instruction and d**
Milenkovic, A.; Milenkovic, M.;
Workload Characterization, 2003. WWC-
Workshop on
27 Oct. 2003 Page(s):99 - 107
Digital Object Identifier 10.1109/WWC.2
AbstractPlus | Full Text: PDF(577 KB)
Rights and Permissions

☐ **10. A measurement study of memory trans**
**PowerPC-based Macintosh**
Adams, T.;
Compcon '96. 'Technologies for the Infor
Papers
25-28 Feb. 1996 Page(s):100 - 110
Digital Object Identifier 10.1109/CMPCC
AbstractPlus | Full Text: PDF(764 KB)
Rights and Permissions

☐ **11. Taking into account access patterns irr**
**address traces**

Hammami, O.;
Southeastcon '95. 'Visualize the Future'.,
26-29 March 1995 Page(s):74 - 77
Digital Object Identifier 10.1109/SECON
AbstractPlus | Full Text: PDF(276 KB)
Rights and Permissions

☐ **12. Multiprocessor cache analysis using A**
Sites, R.L.; Agarwal, A.;
Computer Architecture, 1988. Conferenc
International Symposium on
30 May-2 June 1988 Page(s):186 - 195
Digital Object Identifier 10.1109/ISCA.1
AbstractPlus | Full Text: PDF(872 KB)
Rights and Permissions

☐ **13. Techniques for compressing program a**
Pleszkun, A.R.;
Microarchitecture, 1994. MICRO-27. Pro
International Symposium on
30 Nov.-2 Dec. 1994 Page(s):32 - 39
Digital Object Identifier 10.1109/MICRO
AbstractPlus | Full Text: PDF(796 KB)
Rights and Permissions

☐ **14. Stream-Based Trace Compression**
Milenkovic, A.; Milenkovic, M.;
Computer Architecture Letters, IEEE
Volume 2, Issue 1, Jan. 2003 Page(s):4 -
Digital Object Identifier 10.1109/L-CA.2
AbstractPlus | Full Text: PDF(488 KB)
Rights and Permissions

☐ **15. Automatic and efficient evaluation of n systems**
Abraham, S.G.; Mahlke, S.A.;
Microarchitecture, 1999. MICRO-32. Pro
Symposium on
16-18 Nov. 1999 Page(s):114 - 125
Digital Object Identifier 10.1109/MICRO
AbstractPlus | Full Text: PDF(156 KB)
Rights and Permissions

☐ **16. RECET-a real-time cache evaluation t**
Ahola, J.;
Modeling, Analysis, and Simulation of C
Systems, 1995. MASCOTS '95., Proceedi
Workshop on
18-20 Jan. 1995 Page(s):376 - 381
Digital Object Identifier 10.1109/MASC
AbstractPlus | Full Text: PDF(480 KB)
Rights and Permissions

☐ **17. Compile-time program restructuring i memory systems**
Hartley, S.J.;
Software Engineering, IEEE Transaction
Volume 14, Issue 11, Nov. 1988 Page(s
Digital Object Identifier 10.1109/32.9051

AbstractPlus | Full Text: PDF(460 KB)
Rights and Permissions

☐ **18. Accurate low-cost methods for perform systems**
Laha, S.; Patel, J.H.; Iyer, R.K.;
Computers, IEEE Transactions on
Volume 37, Issue 11, Nov. 1988 Page(s)
Digital Object Identifier 10.1109/12.8699
AbstractPlus | Full Text: PDF(944 KB)
Rights and Permissions

☐ **19. An analysis of cache performance for a**
Stunkel, C.B.; Fuchs, W.K.;
Parallel and Distributed Systems, IEEE T
Volume 3, Issue 4, July 1992 Page(s):42
Digital Object Identifier 10.1109/71.1499
AbstractPlus | Full Text: PDF(1072 KB)
Rights and Permissions

☐ **20. Synthetic traces for trace-driven simul**
Thiebaut, D.; Wolf, J.L.; Stone, H.S.;
Computers, IEEE Transactions on
Volume 41, Issue 4, April 1992 Page(s):
Digital Object Identifier 10.1109/12.1355
AbstractPlus | Full Text: PDF(1584 KB)
Rights and Permissions

☐ **21. The performance of cache-based error**
Janssens, B.; Fuchs, W.K.;
Parallel and Distributed Systems, IEEE T
Volume 5, Issue 10, Oct. 1994 Page(s):1
Digital Object Identifier 10.1109/71.3131
AbstractPlus | Full Text: PDF(788 KB)
Rights and Permissions

☐ **22. Data forwarding in scalable shared-me**
Koufaty, D.A.; Xiangfeng Chen; Poulsen
Parallel and Distributed Systems, IEEE T
Volume 7, Issue 12, Dec. 1996 Page(s):
Digital Object Identifier 10.1109/71.5532
AbstractPlus | References | Full Text: PD
Rights and Permissions

☐ **23. An architecture for tolerating processo multiprocessors**
Banatre, M.; Gefflaut, A.; Joubert, P.; Mc
Computers, IEEE Transactions on
Volume 45, Issue 10, Oct. 1996 Page(s):
Digital Object Identifier 10.1109/12.5437
AbstractPlus | References | Full Text: PD
Rights and Permissions

☐ **24. Multiprocessor memory reference gene**
Rothman, J.B.; Smith, A.J.;
Modeling, Analysis and Simulation of Cc
Systems, 1999. Proceedings. 7th Internati
24-28 Oct. 1999 Page(s):278 - 287
Digital Object Identifier 10.1109/MASC(

AbstractPlus | Full Text: PDF(172 KB)
Rights and Permissions

☐ **25. PDATS II: improved compression of a**
Johnson, E.E.;
Performance, Computing and Communic
IEEE International
10-12 Feb. 1999 Page(s):72 - 78
Digital Object Identifier 10.1109/PCCC.1
AbstractPlus | Full Text: PDF(432 KB)
Rights and Permissions

IEEE Xplore RELEASE 2.1

Welcome United States Patent and Trademark Office

Search Results BROWSE SEARCH IEEE GUID

Results for "(((instruction address execution order)<in>metadata)) <an
1970 <and> ...
Your search matched 0 documents.
A maximum of 100 results are displayed, 25 to a page, sorted by Relevanc
Descending order.

» Search Options

View Session History
New Search

Modify Search

(((instruction address execution order)<in>metadata

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

» Key

IEEE JNL IEEE Journal or Magazine

IEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

**No results were found.**

Please edit your search criteria and try again. Refer assistance revising your search.

Indexed by Inspec

IEEE Xplore RELEASE 2.1

Welcome United States Patent and Trademark Office

Search Results BROWSE SEARCH IEEE GUID

Results for "(((instrument* instruction addresses )<in>metadata)) <and> 1970 <and> p...
Your search matched 0 documents.
A maximum of 100 results are displayed, 25 to a page, sorted by Relevanc Descending order.

» Search Options

View Session History
New Search

Modify Search

(((instrument* instruction addresses )<in>metadata))

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

» Key

IEEE JNL IEEE Journal or Magazine

IEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

**No results were found.**

Please edit your search criteria and try again. Refer assistance revising your search.

Indexed by Inspec